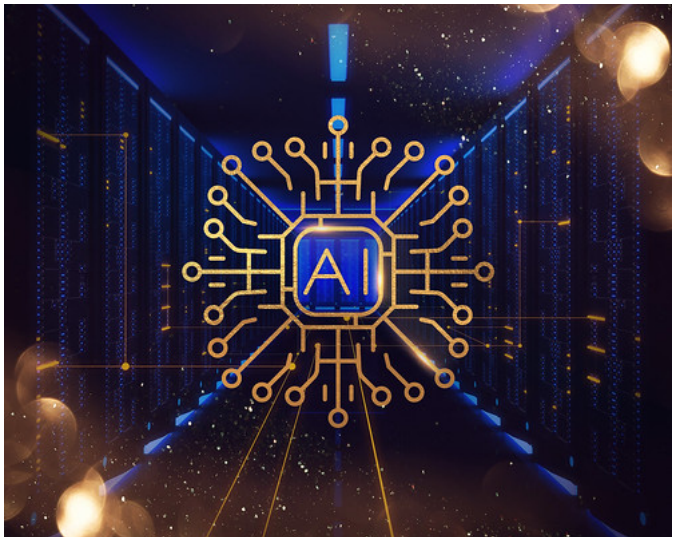


Team Project
Digital Forensics
Design Proposal

June 2023
Version: 1.3

By

GROUP 3

Company: TechNLazy
Industry Focus: Build plugins and agents to handles large volume of files
Founded: August 2022

Disclaimer: *Please note TechNLazy as a company used here is fictitious.*

Version History:

No.	Date	Comments
1.0	19-May-2023	Initial team discussion about planning for the outline
1.1	26-May-2023	Discussing about the approach and assigning tasks to collaborate
1.2	02-Jun-2023	Further discussions and updates on the Risks and Dependencies
1.3	12-Jun-2023	Reviewing and formatting the document

1 Table of Contents

2 Introduction

Digital forensics is a critical function for organisations of all sizes. It investigates and responds to incidents, protects sensitive information, and ensures compliance with legal and regulatory requirements (Pollitt & Zhang, 2016). However, the manual process of searching for file types, archiving them, and analysing the results is time-consuming and error-prone (Nelson, Phillips, & Steuart, 2017). This can hamper an organisation's ability to investigate potential threats and secure critical information efficiently.

This design proposes the automation of file searching, archiving, and transmission for organisations. It uses Python bots and domain-specific knowledge to enhance digital forensics capabilities, streamline investigations, and facilitate the analysis of critical evidence.

This business case presents a solution that automates digital forensics processes within organisations, ensuring compliance with legal and regulatory requirements.

3 Business Requirement

This business case proposes the development of an intelligent agent to address domain-specific digital forensics tasks within an organisation (Casey & Bunting, 2019). The agent will search for specific file types on a file system, archive them, and securely transmit the results for analysis. The proposed solution aims to enhance the organisation's digital forensics capabilities, improve efficiency, and mitigate risks associated with manual data retrieval and analysis (Cohen & Altheide, 2019).

The proposed solution will deliver several benefits for the organisation, including:

- **Improved efficiency:** Automating digital forensics tasks will allow analysts to focus on more complex investigations.
- **Increased accuracy:** The agent's advanced search algorithms and machine learning capabilities will improve the accuracy of investigations by reducing false positives and negatives.
- **Enhanced security:** Secure transmission protocols protect sensitive forensic data from unauthorised access or breaches.
- **Reduced costs:** The agent will help to reduce costs by reducing manual efforts, optimising resource allocation, and minimising human errors.
- **Improved compliance:** The agent's archiving capabilities will help to ensure regulatory compliance by preserving evidence and maintaining the chain of custody.

(Nelson, Phillips, & Steuart, 2017)

The implementation of the proposed solution will be carried out in three phases:

1. **Requirements gathering:** This phase will involve consulting with key stakeholders to understand the organisation's needs and requirements.
2. **Development and testing:** This phase will involve building the intelligent agent, implementing advanced algorithms and encryption protocols, and conducting rigorous testing for functionality and security.
3. **Deployment and integration:** This phase will involve deploying the agent in the organisation's infrastructure and integrating it with file systems, analysis tools, and communication channels.

4 Architecture Overview

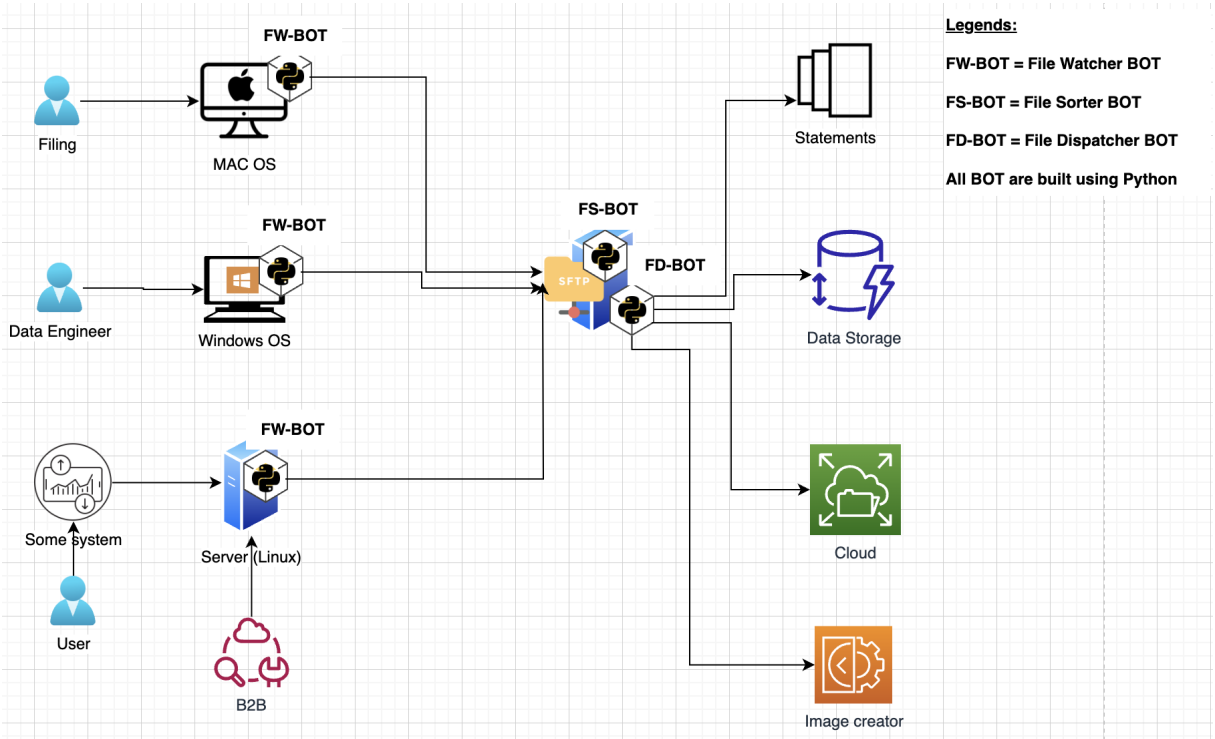
The high-level architecture for the digital forensics agent using Python bots consists of several vital components that work together to perform the required tasks efficiently.

File System Interface:
The agent interacts with the system to search for specific file types using Python's built-in file system APIs or specialised libraries like ***os*** and ***glob***.

Search and Filtering:
The search and filtering utilise Python libraries such as ***re*** (regular expressions) for pattern matching and file type identification.

Archiving:
Files will be archived as required with libraries like ***zipfile*** or ***tarfile*** and then digitally signed with libraries like ***hashlib*** and ***cryptography***.

Secure Communication:
Transmission of files for further analysis uses Python libraries like ***ssl*** or ***paramiko*** for encrypted communication channels to designated destinations.



Logging and Reporting:
Logging the activities of the agents becomes crucial for audit and identifying failures using libraries like ***loguru***.

Future enhancements

The application can be further enhanced by employing the following:

- **Collaboration and Coordination:** Building a multiprocessing capability for agents to carry out tasks with libraries, such as ***multiprocessing*** or ***Celery***.
- **Configuration and Customization:** Enhancing with configuration options to tune the agent's behaviour using libraries like ***configparser*** or frameworks like ***Flask*** for admin tasks.

5 System Requirements

The BOTs are created using Python. The corresponding runtimes for various environments ex; Mac OS, Windows or Linux.

To bundle the application for it to be distributed to various environments we will use PyInstaller.

The minimum python version required would be 3.9.7.

Libraries:

Library	Description	Link/Docs
watchdog	Directory monitoring	https://pythonhosted.org/watchdog/)
shutil	Operation on files	https://docs.python.org/3/library/shutil.html
glob	File pathname pattern and handling	https://docs.python.org/3/library/glob.html
pysftp	Secure file transfer to FTP	https://pypi.org/project/pysftp/
zipfile	Archiving and compression	https://docs.python.org/3/library/zipfile.html
hashlib	Secure Hashing	https://docs.python.org/3/library/hashlib.html
cryptography	Cryptographic recipes	https://pypi.org/project/cryptography/
ssl, paramiko	Secure communication	https://docs.python.org/3/library/ssl.html https://www.paramiko.org/
celery	Queuing and distribution	https://docs.celeryq.dev/en/stable/getting-started/introduction.html
configparser	Config handling	https://docs.celeryq.dev/en/stable/getting-started/introduction.html

6 Design

6.1 Assumptions

- The company has a seasoned workforce and developed procedures for digital forensics.
- Digital forensics is done in the organisation's file systems and data repositories.
- The intelligent agent will be provided with the tools (hardware, software, and processing power) it needs to do its job.
- The intelligent agent will conduct its operations within the company's internal network and be authorised to retrieve, store, and properly send data.
- The architecture assumes the presence of a reliable network infrastructure that allows secure communication between the digital forensics agent and designated destinations.

6.2 Risks

- If the security measures, such as encryption protocols and access controls, are not properly implemented, data breaches or unauthorised access is a danger.
- The intelligent agent may draw the wrong conclusions or overlook crucial evidence because of a high rate of false positives or negatives in the files' search results.
- Challenges in compatibility, performance, or conflicts with other processes may arise while integrating the intelligent agent with pre-existing systems and tools.
- Integration of the agent and its deployment could bring challenges due to incompatibility, conflicts with other components or data.

6.3 Dependencies

- The intelligent agent's capacity to search and archive files depends on the accessibility and compatibility with the organisation's file systems and data repositories.
- To process and interpret the results obtained by the intelligent agent, it must be integrated with existing analysis and reporting technologies used for digital forensics.
- The archived files must be able to be transmitted to chosen analysis destinations using secure communication channels and protocols.

6.4 Approach

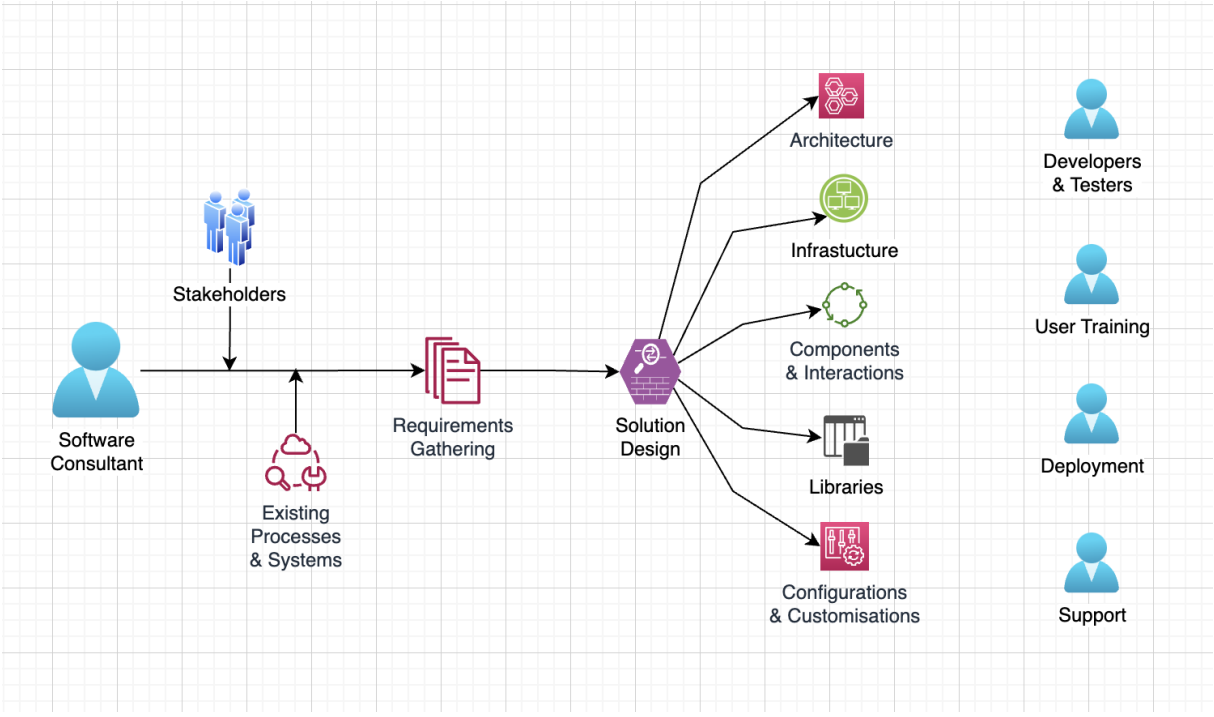
1. **Gathering requirements** entails talking to important people, including digital forensics investigators, to learn what features and capabilities the AI assistant must have.

2. **System architecture design** entails creating a solid framework outlining the intelligent agent's parts, modules, and interfaces. Scalability, adaptability, and compatibility with current infrastructures should all be considered.

3. **Creating an advanced search algorithm** that can accurately identify the desired file kinds by using methods like keyword matching, regular expressions, or machine learning approaches is the work of algorithm developers.

4. **Data confidentiality, integrity, and authenticity** can be guaranteed during archiving and transfer with the help of security measures such as encryption protocols, access controls, and encrypted communication channels.

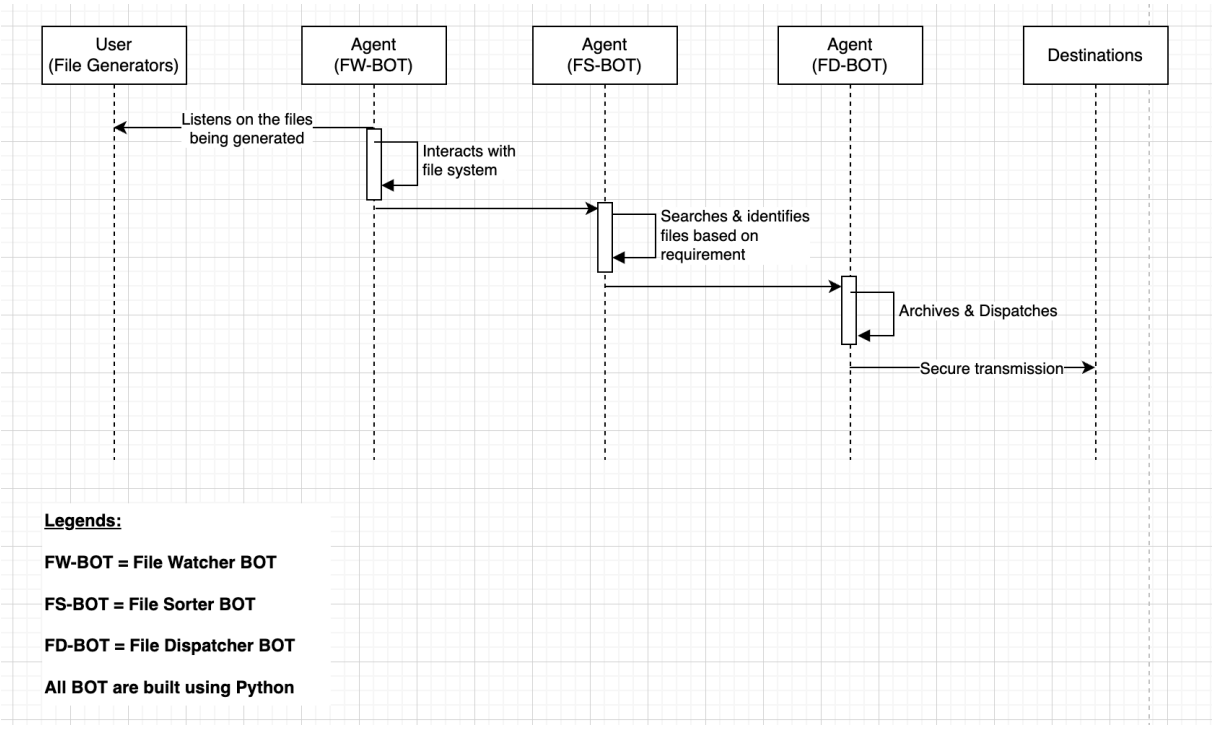
5. The process of connecting the intelligent agent to the company's existing data storage, analysis, and communication mechanisms and **testing** these connections. Validating functionality, performance, and security requires rigorous testing.



6.5 Typical Sequence or Journey

A typical sequence for the BOT to work would be:

1. The bot watches the files to see trigger processes when a file is created.
2. The agent interacts with the file system and applies search and filtering algorithms to identify relevant files.
3. The agent archives and securely transmits the identified files to designated analysis destinations.
4. The analysis destinations perform further analysis and processing of the files.
5. The analysis results are shared with the digital forensics team, who provide feedback to the agent.
6. The agent iteratively refines its search criteria and behaviour based on the feedback.
7. The process may involve multiple iterations until the digital forensics team is satisfied with the results.
8. The digital forensics agent is regularly maintained and updated to remain effective.



6.6 Pattern

The pattern used is the “iterative pattern”, which is a widely recognised pattern in software design. This is based on the following:

1. The intelligent agent should be built modularly, allowing room for expansion and improvement.
2. When rules and best practices for secure coding are followed, vulnerabilities are reduced, and attacks are thwarted.
3. Implementing the repository pattern results in a suitable means of maintaining and accessing data stores and file systems.
4. Using the observer pattern to improve the intelligent agent's inter-module and inter-component communication and notification.
5. The pattern allows to performance of repeated searches, archiving and analysis functionalities, searching for specific file types iteratively, archiving them, transmitting them for analysis, receiving results, and incorporating insights into subsequent iterations.

7 Conclusion

By adopting this architecture, the organisation benefits from reduced manual work. With ongoing enhancements, maintenance, and support, this can provide long-term effectiveness.

The automation capabilities, commitment to legal standards, collaboration features, and ongoing maintenance make it a powerful tool to help organisations improve productivity, enhance evidence handling, and achieve successful outcomes in digital investigations.

This application's capabilities would improve the organisation's digital forensics capabilities.

8 References

- Casey, E., & Bunting, S. (2019). Digital evidence and computer crime: forensic science, computers, and the internet. Academic Press.
- Carrier, B., & Spafford, E. H. (Eds.). (2003). Incident response and computer forensics. O'Reilly Media, Inc.
- Cohen, F., & Altheide, C. (2019). Digital forensics: an introduction. Routledge.
- Nelson, B., Phillips, A., & Steuart, C. (2017). Guide to computer forensics and investigations. Cengage Learning.
- Pollitt, M. M., & Zhang, J. (2016). Digital forensics readiness: Do you have what it takes to survive a cybersecurity incident? Computers & Security, 57, 47-58.